

First Hit   Fwd Refs

Generate Collection

Print

L18: Entry 14 of 44

File: USPT

Nov 13, 2001

US-PAT-NO: 6317820

DOCUMENT-IDENTIFIER: US 6317820 B1

TITLE: Dual-mode VLIW architecture providing a software-controlled varying mix of instruction-level and task-level parallelism

DATE-ISSUED: November 13, 2001

## INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Shiell; Jonathan H.	Dallas	TX		
Bartley; David H.	Dallas	TX		

## ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Texas Instruments Incorporated	Dallas	TX			02

APPL-NO: 09/ 314494   [PALM]

DATE FILED: May 19, 1999

## PARENT-CASE:

This application claims priority under 35 USC .sctn.119(e)(1) of Provisional Application No. 60/088,314, filed Jun. 5, 1998.

INT-CL: [07] G06 F 9/46

US-CL-ISSUED: 712/32; 712/24, 712/32, 712/200, 712/229, 712/245

US-CL-CURRENT: 712/32; 712/200, 712/229, 712/24, 712/245

FIELD-OF-SEARCH: 712/20, 712/24, 712/229, 712/245, 712/246, 712/32, 712/200

PRIOR-ART-DISCLOSED:

## U.S. PATENT DOCUMENTS

Search Selected

Search ALL

Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<input type="checkbox"/>	<u>3573852</u>	April 1971	Watson et al.	709/107
<input type="checkbox"/>	<u>3728692</u>	April 1973	Fennel, Jr.	712/205
<input type="checkbox"/>	<u>3771138</u>	November 1973	Celtruda et al.	712/205
<input type="checkbox"/>	<u>4197579</u>	April 1980	Otis, Jr. et al.	712/214

<input type="checkbox"/>	<u>4229790</u>	October 1980	Gilliland et al	709/101
<input type="checkbox"/>	<u>4320453</u>	March 1982	Roberts et al.	712/246
<input type="checkbox"/>	<u>4344129</u>	August 1982	Asada et al.	712/229
<input type="checkbox"/>	<u>4821187</u>	April 1989	Ueda et al.	712/246
<input type="checkbox"/>	<u>4939638</u>	July 1990	Stephenson et al.	710/244
<input type="checkbox"/>	<u>5530889</u>	June 1996	Kametani	712/247
<input type="checkbox"/>	<u>5539911</u>	July 1996	Nguyen et al.	712/23
<input type="checkbox"/>	<u>5574939</u>	November 1996	Keckler et al.	712/24
<input type="checkbox"/>	<u>5724565</u>	March 1998	Dubey et al.	712/245
<input type="checkbox"/>	<u>5761522</u>	June 1998	Hisanaga et al.	712/1
<input type="checkbox"/>	<u>5848289</u>	December 1998	Studor et al.	712/32
<input type="checkbox"/>	<u>6170051</u>	January 2001	Dowling	712/225

## OTHER PUBLICATIONS

Mudge, Trevor, Strategic Directions in Computer Architecture, ACM Computing Surveys, vol. 28, No. 4, Dec., 1996, pp. 671-678.  
Lo, Jack, et al., Converting Thread-Level Parallelism to Instruction-Level Parallelism via Simultaneous Multithreading, ACM Transactions on Computer Systems, vol. 15, No. 3, Aug., 1997, pp. 322-354.

ART-UNIT: 273

PRIMARY-EXAMINER: Kim; Kenneth S.

ATTY-AGENT-FIRM: Marshall, Jr.; Robert D. Brady, III; W. James Telecky, Jr.; Frederick J.

## ABSTRACT:

This invention is a very long instruction word data processor including plural data registers, plural functional units and plural program counters and is selectively operable in either a first or second mode. In the first mode, the data processor executes a single instruction stream. In the second mode, the data processor executes two independent program instruction streams simultaneously. In the second mode the data processor may respond to two instruction streams accessing only corresponding halves of the data registers and function units. Alternatively, the data processor may respond to a first instruction stream including instructions referencing the whole data processor employing A side function units by alternatively dispatching (1) instructions referencing the A side data registers and the A side function units and (2) instructions referencing the B side data registers and the B side function units. In the first mode, the data processor fetches N bits of instructions each cycle. In the second mode the data processor may fetch N bits of instructions for alternate program counters on alternate cycles or fetches N/2 bits of each of the first and second program counters. The data processor includes interrupt steering and masking control logic allowing instructions to control whether the first instruction stream or the second instruction stream receives interrupts.

9 Claims, 4 Drawing figures

[First Hit](#)   [Fwd Refs](#)

Generate Collection

Print

L18: Entry 14 of 44

File: USPT

Nov 13, 2001

DOCUMENT-IDENTIFIER: US 6317820 B1

TITLE: Dual-mode VLIW architecture providing a software-controlled varying mix of instruction-level and task-level parallelism

Brief Summary Text (6):

This problem has been addressed in a number of ways. One example of the prior art is the VLIW approach to multithreading as shown in U.S. Pat. No. 5,574,939 entitled "MULTIPROCESSOR COUPLING SYSTEM WITH INTEGRATED COMPILE AND RUN TIME SCHEDULING FOR PARALLELISM" by Keckler et. al. Keckler et. al. shows a VLIW system that can execute multiple threads that have been intermixed at compile time into a single VLIW word. In this approach a number of different instruction streams, which would have needed separate program counters, are statically scheduled together and run as a single combined instruction stream under control of a single program counter.

First Hit

Generate Collection

Print

L20: Entry 4 of 10

File: EPAB

Nov 24, 1994

PUB-NO: WO009427216A1

DOCUMENT-IDENTIFIER: WO 9427216 A1

TITLE: MULTIPROCESSOR COUPLING SYSTEM WITH INTEGRATED COMPILE AND RUN TIME  
SCHEDULING FOR PARALLELISM

PUBN-DATE: November 24, 1994

## INVENTOR-INFORMATION:

NAME

COUNTRY

KECKLER, STEPHEN W

DALLY, WILLIAM J

## ASSIGNEE-INFORMATION:

NAME

COUNTRY

MASSACHUSETTS INST TECHNOLOGY

US

APPL-NO: US09404869

APPL-DATE: May 4, 1994

PRIORITY-DATA: US06238893A (May 14, 1993)

INT-CL (IPC): G06F 9/38

EUR-CL (EPC): G06F009/38; G06F009/38, G06F009/38 , G06F009/46

## ABSTRACT:

CHG DATE=19990617 STATUS=O>In a parallel data processing system, very long instruction words (VLIW) define operations able to be executed in parallel. The VLIWs corresponding to plural threads of computation are made available to the processing system simultaneously. Each processing unit pipeline includes a synchronizer stage for selecting one of the plural threads of computation for execution in that unit. The synchronizers allow the plural units to select operations from different thread instruction words such that execution of VLIWs is interleaved across the plural units. The processors are grouped in clusters of processors which share register files. Cluster outputs may be stored directly in register files of other clusters through a cluster switch.

First Hit

Generate Collection

Print

L20: Entry 5 of 10

File: DWPI

Mar 25, 2004

DERWENT-ACC-NO: 2004-073837

DERWENT-WEEK: 200422

COPYRIGHT 2004 DERWENT INFORMATION LTD

TITLE: Programs execution method in cell-phone system, involves splitting instruction words into modified-instruction words, and entering no-operation instructions in modified-instruction words

INVENTOR: BORNEO, A M; PAU, D P ; ROVATI, F S

## PATENT-ASSIGNEE:

ASSIGNEE

CODE

STMICROELECTRONICS SRL

SGSA

BORNEO A M

BORNI

PAU D P

PAUDI

ROVATI F S

ROVAI

PRIORITY-DATA: 2002EP-0425436 (July 2, 2002)

Search Selected

Search ALL

Clear

## PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
<input type="checkbox"/> <u>US 20040059894 A1</u>	March 25, 2004		000	G06F015/00
<input type="checkbox"/> <u>EP 1378824 A1</u>	January 7, 2004	E	018	G06F009/38
<input type="checkbox"/> <u>US 20040054882 A1</u>	March 18, 2004		000	G06F015/00

DESIGNATED-STATES: AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LI LT LU LV  
MC MK NL PT RO SE SI SK TR

## APPLICATION-DATA:

PUB-NO	APPL-DATE	APPL-NO	DESCRIPTOR
US20040059894A1	July 1, 2003	2003US-0612825	
EP 1378824A1	July 2, 2002	2002EP-0425436	
US20040054882A1	July 1, 2003	2003US-0612831	

INT-CL (IPC): G06 F 9/38; G06 F 15/00; G06 F 15/76

ABSTRACTED-PUB-NO: EP 1378824A

EQUIVALENT-ABSTRACTS: NOVELTY - The instructions of programs are compiled as instruction words of given length executable on very long instruction word (VLIW) processor (1). The instruction words are modified into modified-instruction words executable on another VLIW processor (2), by splitting instruction words into modified-instruction words and entering no-operation (nop) instructions in

modified-instruction words. DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for multiprocessor system. USE - For executing programs on multiprocessor system (claimed) in cell-phone system. ADVANTAGE - Enables execution of programs indistinctly on two or more processors and assurance of binary compatibility between the processors, by splitting the instruction words into modified-instruction words. DESCRIPTION OF DRAWING(S) - The figure illustrates the issuing instructions of the program execution process.

CHOSEN-DRAWING: Dwg.9/13

TITLE-TERMS: PROGRAM EXECUTE METHOD CELL TELEPHONE SYSTEM SPLIT INSTRUCTION WORD  
MODIFIED INSTRUCTION WORD ENTER NO OPERATE INSTRUCTION MODIFIED INSTRUCTION WORD

DERWENT-CLASS: T01 W01

EPI-CODES: T01-D02; T01-F03B; W01-C01D3C; W01-C01Q3;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: N2004-059504

First Hit**End of Result Set**☐ **Generate Collection** **Print**

L20: Entry 10 of 10

File: DWPI

Nov 24, 1994

DERWENT-ACC-NO: 1995-007014

DERWENT-WEEK: 199651

COPYRIGHT 2004 DERWENT INFORMATION LTD

TITLE: Multiprocessor coupling system with integrated couple and run time scheduling for parallel operation - has very long instruction words with fields to identify operations able to be performed in parallel for respective threads of computation

INVENTOR: DALLY, W J; KECKLER, S W

PATENT-ASSIGNEE:

ASSIGNEE

CODE

MASSACHUSETTS INST TECHNOLOGY

MASI

PRIORITY-DATA: 1993US-0062388 (May 14, 1993), 1995US-0496462 (June 29, 1995)

**Search Selected****Search ALL****Clear**

## PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
<input type="checkbox"/> <u>WO 9427216 A1</u>	November 24, 1994	E	045	G06F009/38
<input type="checkbox"/> <u>US 5574939 A</u>	November 12, 1996		016	G06F015/16

CITED-DOCUMENTS:4.Jnl.Ref; EP 479390 ; EP 492968

## APPLICATION-DATA:

PUB-NO	APPL-DATE	APPL-NO	DESCRIPTOR
WO 9427216A1	May 4, 1994	1994WO-US04869	
US 5574939A	May 14, 1993	1993US-0062388	Cont of
US 5574939A	June 29, 1995	1995US-0496462	

INT-CL (IPC): G06 F 9/38; G06 F 15/16

ABSTRACTED-PUB-NO: US 5574939A

BASIC-ABSTRACT:

The parallel data processing system has very long instruction words (VLIW) to define operations to be executed in parallel. The VLIWs corresp to number threads of computation are made available to the processing system simultaneously. Each processing unit pipeline includes a synchroniser stage to select one of the number threads of computation for execution in that unit. The synchronisers allow the

number units to select operations from different thread instruction words such that execution of VLIWs is interleaved across the number threads of computation for execution in that unit.

The synchronisers allow the units to select operations from different thread instruction words such that execution of VLIWs is interleaved across the number units. The processors are grouped in clusters of processors which share register files. Cluster outputs may be stored directly in register files off other clusters through a cluster switch.

USE - Scheduling parallel operations using very long instruction words (VLIW) to identify multiple operations to be performed in parallel.

ABSTRACTED-PUB-NO:

WO 9427216A

EQUIVALENT-ABSTRACTS:

A system as claimed in claim 19 wherein the processing units execute an operation which transfers data from one cluster to the register file of another cluster within a thread of computation.

CHOSEN-DRAWING: Dwg.4/6e Dwg.1/6

TITLE-TERMS: MULTIPROCESSOR COUPLE SYSTEM INTEGRATE COUPLE RUN TIME SCHEDULE  
PARALLEL OPERATE LONG INSTRUCTION WORD FIELD IDENTIFY OPERATE ABLE PERFORMANCE  
PARALLEL RESPECTIVE THREAD COMPUTATION

DERWENT-CLASS: T01

EPI-CODES: T01-F03B; T01-M02C;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: N1995-005659



## WEST Search History





DATE: Friday, April 16, 2004

<u>Hide?</u>	<u>Set Name</u>	<u>Query</u>	<u>Hit Count</u>
	<i>DB=EPAB,JPAB,DWPI,TDBD; PLUR=NO; OP=ADJ</i>		
<input type="checkbox"/>	L23	clustered vliw	0
<input type="checkbox"/>	L22	multiple vliw cores	0
<input type="checkbox"/>	L21	multiple vliw cores	0
<input type="checkbox"/>	L20	vliw and L19	10
<input type="checkbox"/>	L19	multiprocessor or multi-processor or (multiple processors)	12016
	<i>DB=PGPB,USPT; PLUR=NO; OP=ADJ</i>		
<input type="checkbox"/>	L18	l3 with vliw	44
<input type="checkbox"/>	L17	clustered vliw	2
<input type="checkbox"/>	L16	variable issue width	10
<input type="checkbox"/>	L15	synchroniz\$5 same L14	4
<input type="checkbox"/>	L14	vliw processors	254
<input type="checkbox"/>	L13	multiple vliw cores	1
<input type="checkbox"/>	L12	multiple vliw processors	0
<input type="checkbox"/>	L11	multiple vliw pipelines	0
<input type="checkbox"/>	L10	vliw same L9	21
<input type="checkbox"/>	L9	multiple pipelines	384
<input type="checkbox"/>	L8	vliw and L7	5
<input type="checkbox"/>	L7	5459798.uref.	21
<input type="checkbox"/>	L6	vliw same L5	4
<input type="checkbox"/>	L5	L3 with chip	892
<input type="checkbox"/>	L4	L3 adj chip	47
<input type="checkbox"/>	L3	multiprocessor or multi-processor or (multiple processors)	22608
<input type="checkbox"/>	L2	vliw and L1	6
<input type="checkbox"/>	L1	pipeline paths	24

END OF SEARCH HISTORY

First Hit   Fwd Refs

Generate Collection

L28: Entry 2 of 10

File: USPT

Sep 30, 2003

US-PAT-NO: 6629232

DOCUMENT-IDENTIFIER: US 6629232 B1

TITLE: Copied register files for data processors having many execution units

DATE-ISSUED: September 30, 2003

## INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Arora; Ken	Los Gatos	CA		
Sharangpani; Harshvardhan	Santa Clara	CA		
Gupta; Rajiv	Los Altos	CA		

## ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Intel Corporation	Santa Clara	CA			02

APPL-NO: 09/ 609911   [PALM]

DATE FILED: July 3, 2000

## PARENT-CASE:

This application is a continuation of U.S. patent application Ser. No. 09/434,070, abandoned, filed Nov. 5, 1999.

INT-CL: [07] G06 F 13/00

US-CL-ISSUED: 712/29; 712/23

US-CL-CURRENT: 712/29; 712/23

FIELD-OF-SEARCH: 712/23, 712/29, 712/24, 711/5, 711/149, 711/150, 711/168

## PRIOR-ART-DISCLOSED:

## U.S. PATENT DOCUMENTS

Search Selected

Search ALL

Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<input type="checkbox"/>	<u>5301340</u>	April 1994	Cook	712/24
<input type="checkbox"/>	<u>5574939</u>	November 1996	Keckler et al.	712/200
<input type="checkbox"/>	<u>5644780</u>	July 1997	Luick	711/127
<input type="checkbox"/>	<u>5826096</u>	October 1998	Baxter	712/215
<input type="checkbox"/>	<u>6219777</u>	April 2001	Inoue	712/10

☐ 6282585

August 2001

Batten et al.

710/20

ART-UNIT: 2183

PRIMARY-EXAMINER: Coleman; Eric

ATTY-AGENT-FIRM: Schwegman, Lundberg, Woessner &amp; Kluth, P.A.

## ABSTRACT:

Interconnect-dominated large register files are reduced in chip area and delay time. A register file in a processor having a number of execution units is divided into multiple copies. Different groups of execution units can read from and write to their own copy of the file registers by a set of local read and write ports. All of the register-file copies are synchronized by writing data from the execution units to remote write ports in at least some registers in other copies of the register file. Each copy can be divided into local and global registers. While all copies of the global registers continue to be written by the remote write ports, the local registers can be written only by a local cluster of execution units. Alternatively or additionally, all of the execution units can write to their local register-file copy, but only some of the units can write the global registers in all copies of the register file.

29 Claims, 7 Drawing figures

First Hit    Fwd Refs☐ Generate Collection

L28: Entry 2 of 10

File: USPT

Sep 30, 2003

DOCUMENT-IDENTIFIER: US 6629232 B1

TITLE: Copied register files for data processors having many execution units

Abstract Text (1):

Interconnect-dominated large register files are reduced in chip area and delay time. A register file in a processor having a number of execution units is divided into multiple copies. Different groups of execution units can read from and write to their own copy of the file registers by a set of local read and write ports. All of the register-file copies are synchronized by writing data from the execution units to remote write ports in at least some registers in other copies of the register file. Each copy can be divided into local and global registers. While all copies of the global registers continue to be written by the remote write ports, the local registers can be written only by a local cluster of execution units. Alternatively or additionally, all of the execution units can write to their local register-file copy, but only some of the units can write the global registers in all copies of the register file.

Brief Summary Text (2):

The present invention relates to electronic data processing, and more specifically concerns an organization for general-purpose register files in superscalar or very long instruction word (VLIW) processor architectures having a large number of execution units connected to the same registers.

Brief Summary Text (5):

One important structure required by all microprocessors is a file of general-purpose or architectural registers. Register files in modern processors, especially those in superscalar, VLIW, and other regularized architectures, are dominated both in timing and in chip area by the metal interconnections required for data and address lines. This situation becomes even worse because of the increasing parallel-execution width of present and future designs--because of the larger number of instructions that can be executed in parallel. The importance of interconnect area and delay in large regular structures such as register files has not been appreciated in the past.

Brief Summary Text (9):

The invention employs multiple copies of a register file in a processor having a number of execution units that access the register file. Each group of execution units can read from and write to its own copy of the file registers by a set of local read and write ports. In addition, all of the register-file copies are synchronized by writing data to remote write ports in the other copies of the register file. The interconnections between the execution units and the register-file copies thus grow less rapidly than they otherwise would, and the difference becomes greater as the execution width of the machine increases.

Detailed Description Text (3):

FIG. 1 is a high-level block diagram of a typical superscalar or VLIW processor 100. Memory 110, which can include on-chip cache memory, off-chip cache, system memory, and even storage devices such as disk drives, couples to an instruction decoder 120 and to a number of parallel execution units 130. The term "execution unit" must be given a broad meaning; anything that sends data to and/or receives

data from the register file will profit from the invention.

Detailed Description Text (9):

Writing data to registers is the only operation by which the two register-file copies could possibly ever become unsynchronized with each other. External data is written to both copies in parallel, and read operations by units 130 do not alter the contents of a register. The present invention provides execution units 131 with another set of interconnections 335, called remote write connections, that lead to the other copy 320 of the register file, for the purpose of synchronizing write operations between the two copies. While local write connections 333 present data and addresses to the local register-file copy 310, connections 335 present the same full set of addresses and data in parallel to the remote register-file copy 320. Remote write connections 336 serve the same purpose where register-file copy 320 is the local copy and register-file copy 310 is the remote. That is, remote write connections 335 and 336 preserve the synchrony of both register-file copies, so that each contains exactly the same data at all times.

US Reference Patent Number (2):

5574939

[First Hit](#)   [Fwd Refs](#)☐ **Generate Collection**

L28: Entry 3 of 10

File: USPT

Jun 4, 2002

US-PAT-NO: 6401190

DOCUMENT-IDENTIFIER: US 6401190 B1

**\*\* See image for Certificate of Correction \*\***

TITLE: Parallel computing units having special registers storing large bit widths

DATE-ISSUED: June 4, 2002

## INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Nishioka; Kiyokazu	Odawara			JP
Tanaka; Kazuhiko	Fujisawa			JP
Fujikawa; Yoshifumi	Yokohama			JP
Nojiri; Toru	Kawasaki			JP
Kojima; Keiji	Sagamihara			JP
Terada; Koichi	Yokohama			JP
Kurokawa; Yoshiki	Chigasaki			JP
Hosoki; Koji	Yokohama			JP

## ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Hitachi, Ltd.	Tokyo			JP	03

APPL-NO: 08/ 913840   [PALM]

DATE FILED: September 12, 1997

## FOREIGN-APPL-PRIORITY-DATA:

COUNTRY	APPL-NO	APPL-DATE
JP	7-058790	March 17, 1995

## PCT-DATA:

APPL-NO	DATE-FILED	PUB-NO	PUB-DATE	371-DATE	102(E)-DATE
PCT/JP96/00673	March 15, 1996	WO96/29646	Sep 26, 1996	Sep 12, 1997	Sep 12, 1997

INT-CL: [07] G06 F 15/00

US-CL-ISSUED: 712/24

US-CL-CURRENT: 712/24

FIELD-OF-SEARCH: 712/22, 712/23, 712/24

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected

Search ALL

Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<input type="checkbox"/>	<u>5057837</u>	October 1991	Colwell et al.	341/55
<input type="checkbox"/>	<u>5301340</u>	April 1994	Cook	395/800
<input type="checkbox"/>	<u>5513363</u>	April 1996	Kumar et al.	395/800
<input type="checkbox"/>	<u>5530817</u>	June 1996	Masubuchi	395/375
<input type="checkbox"/>	<u>5574939</u>	November 1996	Keckler et al.	395/800
<input type="checkbox"/>	<u>5870618</u>	February 1999	Fujikawa et al.	395/800.01

## FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
1-133138	May 1989	JP	
4-308930	October 1992	JP	
5-143333	June 1993	JP	
5-233281	September 1993	JP	
63-97833	April 1998	JP	

## OTHER PUBLICATIONS

Steven et al., "iHARP: a multiple instruction issue processor", IEE Proceedings, pp. 439-449, 1992.\*  
Findlay et al., "HARP: A VLIW RISC Processor", IEEE, pp. 368-372, 1991.

ART-UNIT: 2154

PRIMARY-EXAMINER: Follansbee; John A.

ATTY-AGENT-FIRM: Kenyon &amp; Kenyon

## ABSTRACT:

An object of the prevent invention is to provide a processor that can execute many computations with a small number of instruction codes. As far as multimedia processing is concerned, a plurality of computations of a same type are often executed concurrently and hence a plurality of computing units having a same function are used and mode information for controlling the plurality of units by an instruction unit for one computing unit is prepared in each instruction to execute a plurality of computations with a single instruction.

2 Claims, 30 Drawing figures

First Hit    Fwd Refs☐ Generate Collection

L28: Entry 3 of 10

File: USPT

Jun 4, 2002

DOCUMENT-IDENTIFIER: US 6401190 B1

**\*\* See image for Certificate of Correction \*\***

TITLE: Parallel computing units having special registers storing large bit widths

Brief Summary Text (6):

Approaches for good use of a plurality of computing units include superscalar architecture and VLIW (Very Long Instruction Word). The former is mainly used by general-purpose processors and the scheduling for concurrently executing a plurality of computational operations is performed by these processors. This approach is advantageous in exchangeability of objects with an existing single-processing processor, but at the cost of its extremely complicated hardware because the scheduling is dynamically performed by the processors. On the other hand, VLIW has a problem of compatibility with existing processors but is advantageous in its simplified hardware because no instruction decoder is required.

Brief Summary Text (7):

One of the points of the VLIW hardware simplification is its instruction format. This instruction format is composed of fields for directly controlling computing units, thereby extremely simplifying the control by hardware. A processor having such an instruction format is disclosed in Japanese Non-examined Patent Publication No. Sho 63-98733 "COMPUTER CIRCUIT CONTROL METHOD" for example. In this citation, an operation field indicating that a micro instruction for computation is an instruction for computation and a plurality of control bits for controlling a computing circuit are provided, directly controlling each part of the computing circuit by each of these control bits. Thus, VLIW can implement parallel processing by comparatively simple hardware.

Brief Summary Text (8):

As described, superscalar architecture and VLIW provide effective means for enhancing processing parallelism to draw out performance. In order to fully draw out parallelism, the help of a compiler is indispensable. To be specific, a technique such as loop expansion is known. In this technique, a loop body in a program is duplicated (expanded) a plurality of times and the codes in the expanded loop are scheduled. Namely, increasing the number of instructions to be executed between loop return branches increases the possibility of executing a plurality of instructions concurrently.

Brief Summary Text (13):

Still another object of the present invention is to provide a VLIW processor based on static scheduling.

Brief Summary Text (14):

Yet another object of the present invention is to provide a VLIW processor compatible with various applications and enhanced in the operating ratios of the computing units.

Brief Summary Text (20):

For example, in order to execute a plurality of computations with a single instruction by a plurality of computing devices, in a VLIW processor in which one instruction is constituted by a plurality of fields for controlling the computing



devices, mode information for controlling the plurality of computing devices is provided in one field. Further, an instruction expansion circuit for generating a plurality of fields from one field in one instruction is provided and the above-mentioned plurality of computing devices are constituted by arranging a plurality of computing devices having a same function.

Brief Summary Text (22):

In a processor having three or more computing devices, specification information for specifying the computing devices to be executed concurrently is provided and the above-mentioned instruction expansion circuit is provided with a function for generating the required number of instruction fields for the VLIW processor and generating an instruction for the superscalar processor according to the above-mentioned specification information.

Brief Summary Text (49):

According to the present invention, if a VLIW processor has eight computing devices, one instruction is constituted by eight fields. One field has operation information, operand information, and the above-mentioned mode information. If this mode information specifies concurrent computation mode for controlling the plurality of computing devices, the remaining seven fields do not exist in the memory at reading an instruction. Consequently, the instruction expansion circuit copies the operation information and the operand information specified in the above-mentioned one field to generate the remaining seven fields. Thus, one instruction equivalent to eight fields is generated with a code size for one field. Because all computing devices have the same function, a plurality of computation instructions become executable in parallel without problem, resulting in the code size compressed to 1/8. Especially, if computing device specification information is set to the mode information, only the field corresponding to this setting information is generated, so that, if the setting information is provided in three bits, the number of concurrent computations can be controlled in a range of two to eight.

Drawing Description Text (22):

FIG. 21 is a block diagram illustrating a synchronizer 210 in detail.

Detailed Description Text (2):

In what follows, the present invention will be described. FIG. 1 is a block diagram illustrating a VLIW processor to which the present invention is applied. In the figure, reference numeral 1 denotes an instruction memory for storing a compressed instruction, reference numeral 2 denotes an instruction expansion circuit, a main block of the present invention, for expanding a compressed instruction code read from the instruction memory 1 into an actually executable code, reference numeral 3 denotes an address bus of the instruction memory 1, reference numeral 4 denotes a data bus of the instruction memory 1, reference numerals 5 through 12 denote field buses to which the instruction expansion circuit 2 output an expanded code, reference numerals 14 through 21 denote instruction registers for holding expanded codes transferred via the field buses 5 through 12, reference numerals 22 through 25 denote computing unit having a same constitution for executing various computational operations according to the expanded codes held in the instruction registers 14 through 21, reference numeral 26 denotes an IFG (Integer Floating Graphics) computing device for executing complicated computational operations such as a multimedia computation operation for which a plurality of operations are performed on an 8-bit or 16-bit basis and a multiplication, reference numeral 27 denotes an INT (Integer) computing device for executing simple computational operations such as a data transfer instruction for executing data transfer between a data memory 30 and a register file and a logic operation, reference numeral 28 denotes a register file for holding a value to be operated and an operation result value, composed of 32 64-bit registers, and having 4 read ports and 3 write ports, reference numeral 29 denotes a selection circuit for transferring operation results of the computing units 22 through 25 to another operation unit, and reference

numeral 30 denotes the data memory with which data is transferred with the register files in the computing units 22 through 25.

Detailed Description Text (3):

In this figure, this VLIW processor is formed on a single LSI. Descriptions of a cache memory for temporarily storing instruction codes and so on and LSI terminals for reading instruction codes and so on from outside the processor and outputting operation results to the outside are omitted from the following description.

Detailed Description Text (19):

In the normal mode ("S mode" is 0), an operation result of each computing unit can be written to a register of the register file in another computing unit. Therefore, in the normal mode, a computing unit is identified by "dest. bank" and a register in that computing unit is identified by a destination block (hereafter referred to as "destination") composed of bits 12 through 16. The computing unit 22 corresponds to bank 0, the computing unit 23 corresponds to bank 1, the computing unit 24 corresponds to bank 2, and the computing unit 25 corresponds to bank 3. The "destination" can specify 32 register numbers and the dest. bank can specify 8 computing units. The present embodiment is constituted by the four computing units 22 through 25 but the instruction format itself is applicable to a VLIW processor constituted by eight computing units.

Detailed Description Text (142):

Normally, in VLIW, about 80% of the objects is occupied by NOP. Therefore, NOP compression is an essential technology when memory usage efficiency is taken into consideration. Use of the header used by this technology also in the SIMD mode mitigates the overhead, which is the feature of this embodiment.

Detailed Description Text (143):

In the above-mentioned embodiment, the SIMD mode is implemented by adding four bits to each field. If no header is assumed, it is necessary to add seven bits to each field for implementing the SIMD mode. Namely, in addition to the four bits used in this embodiment, two bits for field address and one bit for synchronization control are required.

Detailed Description Text (144):

Because an omitted field exists in the SIMD mode, each field needs to know the fields 0, 1, fields 2, 3, fields 4, 5, or the fields 6, 7. Necessary for this are the two bits for field address. In addition, because the number of fields of one instruction is not constant, distinction between instructions is not known. To make the distinction clear, the 1-bit of synchronization control is required. By inverting information of this bit for every instruction, the distinction can be detected. Therefore, if compression in unit of one field is considered like the embodiment, the following number of bits are required for one instruction (32 bits.times.8=256 bits):

Detailed Description Text (148):

The following describes a method not assuming the above-mentioned header practiced as a second embodiment of the invention with reference to FIG. 17. Especially, in the second embodiment, the above-mentioned compression in unit of two fields is considered. FIG. 17 is a block diagram illustrating a VLIW processing in its entirety. In the figure, the circuit blocks and signal lines similar to those previously described with FIG. 1 are denoted by the same reference numerals. In the figure, reference numeral 200 denotes an instruction expansion circuit different from that shown in FIG. 1. In the present embodiment, no header is used, so that one instruction always consists of 32 bytes or less and the refetch signal line 13 of FIG. 1 is not required. Namely, EXP2 stage required by the instruction 5 shown in FIG. 10 does not exist. This is one of features of the present embodiment. Except for this point and the internal operations of the instruction expansion circuit 200, the present embodiment is the same as the embodiment of FIG. 1.

Detailed Description Text (154):

In the figure, reference numeral 210 denotes a synchronizer for generating, from the information of the compressed field bus 41 and the instruction address bus 64, information to be outputted to the instruction length signal line 68 and the write enable bus 43, and reference numeral 211 denotes a select signal generator for generating, from the information of the compressed field bus 41 and the write enable bus 43, select information to be outputted to select information lines 206 through 209. The address controller 61 is basically the same in function as the address controller 61 shown in FIG. 8.

Detailed Description Text (155):

The synchronizer 210 inputs sync bits 41a, c, e, and g from the compressed field bus 41. And, by inputting an instruction address from the instruction address bus 64, it can be known to which sync bit the instruction in execution corresponds. Further, by checking the change point of the sync bit, the length of the instruction can be known. Still further, the synchronizer identifies data in the above-mentioned compressed field bus in which the instruction exists and then outputs information for indicate the position in the instruction buffer 40 to which the data is written to the instruction length signal line 68.

Detailed Description Text (156):

The select signal generator 211 receives information of the write enable bus 43, 41a, c, e, and g of "SIMD" and "S mode" and address information from the compressed field bus 41. From these pieces of information, the select signal generator outputs four bits of positional information (information indicating one of the four bits 41a, c, e, and h) of the field 0 to the select information line 206. If the field 0 is NOP-compressed, all four bits go 0. This is, at the same time, provides the select information of the field 1 (information indicating one of the four bits 41b, d, f, and g). Likewise, the select signal generator outputs four bits of positional information (information indicating one of 41a, c, e, and h) of the field 2 to the select information line 207, four bits of positional information (information indicating one of 41a, c, e, and h) of the field 4 to the select information line 208, and four bits of positional information (information indicating one of 41a, c, e, and h) of the field 6 to the select information line 209. The following describes detailed operations of the synchronizer 210 and the select signal generator 211.

Detailed Description Text (157):

FIG. 21 shows a block diagram illustrating the synchronizer 210 in detail. In the figure, the circuit blocks and signal lines similar to those previously described with FIG. 20 are denoted by the same reference numerals.

Detailed Description Text (179):

The above-mentioned embodiments 1, 2, and 3 are applied to a VLIW processor that presupposes static scheduling, but the present invention is not necessarily limited thereto. For example, the present invention is also applicable to a superscalar processor that performs dynamic scheduling. On instruction in the superscalar processor is basically constituted by a fixed length of one field as described in the preceding embodiment. Such a processor incorporates a plurality of computing units and an instruction queues and has a dispatcher that checks the dependent relationship between the plurality of queued instructions and, if no dependency is found and a plurality of executable instructions are found, transfers these instructions to the plurality of computing units simultaneously. Therefore, as shown in FIG. 2 of the present invention, if the SIMD mode is specified in the instruction format by "S-mode" and "SIMD," the above-mentioned dispatch unit transfers that instruction to the plurality of computing units, easily implementing the SIMD mode of the superscalar processor.

Detailed Description Text (201):

As described, the 16 8-bit multipliers occupying most of the circuitry can be shared by normal multiplication instructions and divided multiplication instructions. Arranging a plurality of computing units constituted by the above-mentioned computing devices makes the processor compatible with various applications, thereby implementing a VLIW processor with enhanced availability of each of the computing units constituting the processor.

Detailed Description Text (205):

In addition, the present invention is applicable to processors of various architectures such as VLIW and superscalar.

US Reference Patent Number (5):

5574939

Other Reference Publication (2):

Findlay et al., "HARP: A VLIW RISC Processor", IEEE, pp. 368-372, 1991.

## WEST Search History





DATE: Friday, April 16, 2004

<u>Hide?</u>	<u>Set Name</u>	<u>Query</u>	<u>Hit Count</u>
	<i>DB=USPT; PLUR=NO; OP=ADJ</i>		
<input type="checkbox"/>	L28	synchroniz\$5 and L26	10
<input type="checkbox"/>	L27	synchroniz\$5 and L26	0
<input type="checkbox"/>	L26	vliw and L25	29
<input type="checkbox"/>	L25	5574939.uref.	53
	<i>DB=EPAB,JPAB,DWPI,TDBD; PLUR=NO; OP=ADJ</i>		
<input type="checkbox"/>	L24	5574939.uref.	0
<input type="checkbox"/>	L23	clustered vliw	0
<input type="checkbox"/>	L22	multiple vliw cores	0
<input type="checkbox"/>	L21	multiple vliw cores	0
<input type="checkbox"/>	L20	vliw and L19	10
<input type="checkbox"/>	L19	multiprocessor or multi-processor or (multiple processors)	12016
	<i>DB=PGPB,USPT; PLUR=NO; OP=ADJ</i>		
<input type="checkbox"/>	L18	l3 with vliw	44
<input type="checkbox"/>	L17	clustered vliw	2
<input type="checkbox"/>	L16	variable issue width	10
<input type="checkbox"/>	L15	synchroniz\$5 same L14	4
<input type="checkbox"/>	L14	vliw processors	254
<input type="checkbox"/>	L13	multiple vliw cores	1
<input type="checkbox"/>	L12	multiple vliw processors	0
<input type="checkbox"/>	L11	multiple vliw pipelines	0
<input type="checkbox"/>	L10	vliw same L9	21
<input type="checkbox"/>	L9	multiple pipelines	384
<input type="checkbox"/>	L8	vliw and L7	5
<input type="checkbox"/>	L7	5459798.uref.	21
<input type="checkbox"/>	L6	vliw same L5	4
<input type="checkbox"/>	L5	L3 with chip	892
<input type="checkbox"/>	L4	L3 adj chip	47
<input type="checkbox"/>	L3	multiprocessor or multi-processor or (multiple processors)	22608
<input type="checkbox"/>	L2	vliw and L1	6
<input type="checkbox"/>	L1	pipeline paths	24

END OF SEARCH HISTORY